Term Proximity Scoring for Ad-Hoc Retrieval on Very Large Text Collections

Stefan Büttcher Charles L. A. Clarke Brad Lushman

School of Computer Science
University of Waterloo, Canada

Categories and Subject Descriptors

H.2.4 [**Systems**]: Textual databases; H.3.4 [**Systems and Software**]: Performance evaluation

General Terms

Experimentation, Performance

Keywords

Information Retrieval, Query Processing, Term Proximity

1. INTRODUCTION

Document retrieval functions based on the vector space model, such as Okapi BM25 [2] [3], have been shown to be highly effective in ad-hoc information retrieval tasks. One of their shortcomings is that their bag-of-words approach does not take the proximity of query terms within a document into account and consequently gives the same score to a document regardless whether the query terms appear close to each other within that document or far apart. This contradicts the intuitive understanding that, in a relevant document, query terms appear relatively close to each other and not in completely unrelated parts of the document.

Rasolofo and Savoy [1] were able to show that integrating term proximity into existing vector space retrieval methods can improve the quality of the search results significantly. While trying to reproduce their results on different text collections and to find new ways of integrating term proximity into vector-space-based retrieval functions, we found that term proximity only improves the quality of the search results on some text collections, while it leaves the search system's retrieval effectiveness unaffected on others, or even causes a slight deterioration. Two of the text collections we used in our experiments were the TREC45-CR collection (TREC disks 4&5, without the Congressional Record), consisting of 528,000 documents with an average length of 561 tokens, and the GOV2 collection used in the TREC Terabyte track, consisting of 25.2 million documents with an average length of 1,721 tokens. We found that, while the use of term proximity improved the retrieval effectiveness significantly for GOV2 (paired t-test: $p < 0.02$ for P@10; $p < 0.01$ for P@20), the smaller TREC45-CR collection seemed unimpressed efforts and did not divulge more relevant documents to our term-proximity-based retrieval method than to plain Okapi BM25 (details in Table 1).

	TREC45-CR	GOV2
Collection size (#docs)	528,155	25,205,204
Avg. doc. length (terms)	561	1,721
P@10 (BM25/BM25TP)	0.382/0.381	0.529/0.600
P@20 (BM25/BM25TP)	0.308/0.309	0.494/0.561

Table 1: Collection characteristics and retrieval effectiveness for TREC45-CR and GOV2. Effectiveness for BM25 and BM25TP (BM25 + proximity) was evaluated using 100 topics from the TREC 2003 Robust track (TREC45-CR) and 50 topics from the TREC 2004 Terabyte track (GOV2), respectively.

Based on the characteristics of the collections, this lead us to two hypotheses:

1. Term proximity is more important when the search engine is dealing with longer documents.
2. Term proximity becomes more important as the size of the text collection increases.

We performed additional experiments, with the goal of validating (or refuting) these two theories. We present an experimental evaluation that supports the second hypothesis. For the first hypothesis, no such support could be found. We also show that term proximity is more important for stemmed queries than for unstemmed queries, an aspect that we had ignored in our initial experiments.

2. COMBINING PROXIMITY AND BM25

Given a query containing the terms $T_1, T_2, \ldots, T_n$ the BM25 relevance score of a document D is:

$$\text{Score}_{\text{BM25}}(D) = \sum_{i=1}^{n} w_{T_i} \cdot \frac{f_{D,T_i} \cdot (k_1 + 1)}{f_{D,T_i} + K},$$

$$K = k_1 \cdot (1 - b + \frac{b \cdot |D|}{avgdl}),$$

where f_{D,T_i} is the number of occurrences of the term T_i within D, $|D|$ is the length of D (number of terms), $avgdl$ is the average document length in the collection, and w_{T_i} is T_i's IDF weight: $w_{T_i} = \log \frac{N}{N_{T_i}}$, where N is the total

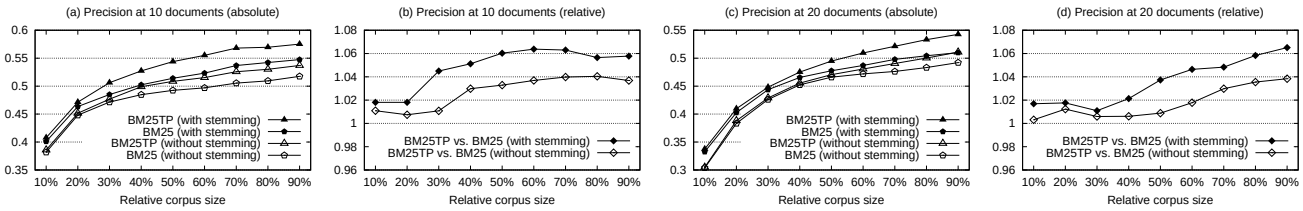

Figure 1: Precision after 10 and after 20 documents – for plain Okapi BM25 and the proximity-enhanced BM25TP, with stemming either turned on or off.

number of documents in the text collection, and N_{T_i} is the number of documents containing T_i. Although there are other variants of the BM25 function, this is the one that is used by our retrieval system.

Our integration of term proximity into the BM25 scoring function is very similar to that presented by Rasolofo and Savoy [1]. For the presentation in this paper, we chose to use our own method instead of theirs because it exhibited slightly better retrieval effectiveness in almost all of our experiments.

Suppose a user submits the query $\mathcal{Q} = \{T_1, \ldots, T_n\}$. Then our implementation of BM25 fetches the posting lists for all query terms from the index and arranges them in a priority queue. It then starts consuming postings from all posting lists, one posting at a time, in ascending order, to find matching documents and simultaneously compute the relevance scores of all matching documents found (document-at-a-time approach). If an index with full positional information is used, Term proximity can be integrated into this process without much effort. With every query term, we associate an accumulator that contains that term's proximity score within the current document. Whenever the search system encounters a posting that belongs to the query term T_j, it looks at the previous posting, belonging to the query term T_k, and determines the distance (number of postings) between the current posting and the previous one. If $T_j \neq T_k$, then both terms' accumulators are incremented:

$$\begin{aligned} acc(T_j) &:= acc(T_j) + w_{T_k} \cdot (dist(T_j + T_k))^{-2}, \\ acc(T_k) &:= acc(T_k) + w_{T_j} \cdot (dist(T_j + T_k))^{-2}, \end{aligned}$$

where w_{T_i} is T_i's IDF weight (cf. equation 1). For $T_j = T_k$, the accumulators remain unchanged. When the end of the current document is reached, the document's score is computed, and all proximity accumulators are reset to zero. The score of a document D is:

$$\begin{aligned} &\text{Score}_{\text{BM25TP}}(D) \\ = \; &\text{Score}_{\text{BM25}}(D) + \sum_{T \in \mathcal{Q}} \min\{1, w_T\} \cdot \frac{acc(T) \cdot (k_1 + 1)}{acc(T) + K}, \end{aligned}$$

where k_1 and K are are the same as in the original Okapi equation. The difference to the strategy followed by Rasolofo and Savoy is that in our approach only neighboring query term's can affect each other's accumulator and that the impact of term proximity on the document score is smaller because the term weight w_T is limited to 1.

3. EXPERIMENTAL RESULTS

For our experiments, we split up the GOV2 collection into 100 random chunks, containing 252,000 documents each. From these chunk, we built subcollections containing 10%, 20%, ..., 90% of the documents in the whole collection. For each size, we constructed 20 such subcollections by randomly picking an appropriate number of chunks and combining them. We then fed 100 queries from the 2004 and 2005 TREC Terabyte ad-hoc retrieval tasks into our system and executed them using different system configurations (BM25 with or without term proximity; stemming enabled or disabled). Note that this is different from our initial experiments, where we only used the 50 topics from 2004. The results depicted in Figure 1 represent the mean precision values over all subcollections of the respective size. The figure shows that the relative gain achieved by BM25TP, compared to the original BM25, increases as the underlying text collection grows. This observation is true for both P@10 and P@20. It also shows that the relative gain is greater for stemmed queries than for unstemmed queries. We surmise that this is because term proximity helps distinguish between stem-equivalent terms that represent the same semantic concept and stem-equivalent terms that don't. We also performed experiments for subcollections of GOV2 containing documents of different average length. However, the results obtained did not indicate any correlation between average document length and the effectiveness of term proximity scoring.

Our explanation of the fact that term proximity is more important for bigger text collections is that for larger collections the likelihood of finding non-relevant documents that contain the query terms by chance is greater than for smaller collections. Term proximity, as an additional feature, helps distinguish between these documents and documents that are actually relevant. An important implication of this finding is that, although using a document-level index instead of a positional index can reduce both time and space complexity greatly when dealing with very large text collections, it is advisable to keep full positional information, as this can significantly increase the system's retrieval effectiveness.

4. REFERENCES

[1] Y. Rasolofo and J. Savoy. Term Proximity Scoring for Keyword-Based Retrieval Systems. In *Proceedings of the 25th European Conference on IR Research (ECIR 2003)*, pages 207–218, April 2003.

[2] S. E. Robertson, S. Walker, and M. Hancock-Beaulieu. Okapi at TREC-7. In *Proceedings of the Seventh Text REtrieval Conference*, Gaithersburg, USA, November 1998.

[3] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference*, Gaithersburg, USA, November 1994.